

Variable	Mean	SD	Min	Max
Age	34.5	10.2	22	55
Gender	Male	Female		
Marital status	Married	Single		
Education	High school	College		
Occupation	Manager	Worker		
Income	\$10,000	\$20,000		
Health status	Good	Fair		
Exercise frequency	Weekly	Monthly		
Stress level	Low	High		
Sleep quality	Good	Poor		
Dietary habits	Healthy	Unhealthy		
Alcohol consumption	None	Occasional		
Tobacco use	Non-smoker	Smoker		
Family size	2	3		
Work hours	40	50		
Commuting time	30	45		
Home ownership	Owner	Renter		
Neighborhood safety	Safe	Unsafe		
Access to green spaces	Yes	No		
Proximity to public transport	Close	Far		
Local economy	Strong	Weak		
Community engagement	High	Low		
Local government effectiveness	Good	Poor		
Local infrastructure quality	High	Low		
Local environmental quality	Good	Poor		
Local social cohesion	High	Low		
Local cultural diversity	High	Low		
Local economic diversity	High	Low		
Local political participation	High	Low		
Local social capital	High	Low		
Local trust in government	High	Low		
Local civic engagement	High	Low		
Local community development	High	Low		
Local social services	Good	Poor		
Local healthcare access	Good	Poor		
Local education quality	Good	Poor		
Local employment opportunities	Good	Poor		
Local income inequality	Low	High		
Local social inequality	Low	High		
Local environmental inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low	High		
Local social inequality	Low	High		
Local economic inequality	Low	High		
Local political inequality	Low			

receiving a problem statement from the applications module;

determining an expected problem solver behavior associated with said configuration parameters for said problem statement;

providing the solving module with configuration parameters;

selecting a set of configuration parameter vectors;

determining a set of search space points;

perform a partial search with said configuration parameter vectors;

determining actual solver behavior;

reviewing said actual solver behavior to determine if a problem solution has been found;

determining whether to perform a solver iteration step or to request the complexity module to perform an adaptation step if a problem solution has not been found;

performing a said solver iteration step when said solver iteration step is selected, comprising the steps of determining new actual solver behavior and determining whether to repeat said solver iteration step;

repeating said solver iteration step until said adaptation step is selected;

comparing said actual solver behavior with said expected solver behavior when said adaptation step is selected;

requesting the complexity module to perform said adaptation step;

performing said adaptation step, comprising the steps of modifying said configuration parameters within the complexity module, configuring the solving module with said modified configuration parameters, determining expected solver behavior associated with said modified configuration parameters for said problem statement, selecting an algorithm to calculate a revised problem solution, determining a revised actual solver behavior associated with said modified configuration parameters for said problem statement, reviewing said revised actual solver behavior to determine if a problem solution has been found, determining whether to perform said solver iteration step or to request the complexity module to perform another adaptation step if a problem solution has not been found, and repeating said iteration step until said adaptation step is selected;

repeating said adaptation step until a problem solution is found; and

providing the solution to the applications module.

2. The problem solving method according to claim 1, further comprising the step of selecting an algorithm to calculate an initial problem solution.

3. The problem solving method according to claim 1, further comprising the step of refining the configuration parameters provided to said solving model.

4. The problem solving method according to claim 1, wherein the problem solver comprises an adaptive constraint problem solver.

5. The problem solving method according to claim 1, further comprising the step of transforming said problem statement after receiving said problem statement from the applications module.

6. The problem solving method according to claim 5, further comprising the step of transforming said problem solution before providing said problem solution to the applications module.

7. The problem solving method according to claim 1, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

8. The problem solving method according to claim 7, further comprising the step of transforming said problem configuration parameters before providing said problem configuration parameters to the solving module.

9. The problem solving method according to claim 1, wherein the step of selecting a set of configuration parameter vectors further comprises:

choosing a set of default configuration parameter vectors;

selecting an initial minimum point;

performing a local search;

evaluating actual behavior to determine whether to repeat a local search or select a different solver algorithm;

repeating a local search with a second minimum point when the step of repeating a local search is selected; and

revising the set of configuration parameter vectors for each search performed.

10. A computer system for problem solving, the system having implementation units communicating with the computer system, the system comprising:

an input device for providing the problem statement;

a computer coupled to the output of said input device;

a memory portion coupled to the computer comprising:

software for receiving the problem statement from said input device;

software for determining solver configuration parameter vectors;

software for configuring a problem solver;

software for determining expected solver behavior;

software for performing a partial search with said configuration parameter vectors;

software for determining actual solver behavior and determining whether a solution has been found;

software for determining whether to perform a solver iteration step or to perform an adaptation step; and

software for performing an adaptation step, comprising modifying said configuration parameters and reconfiguring said problem solver; and

output means for providing a solution statement.

11. The computer system for problem solving according to claim 10, wherein said problem solver comprises an adaptive constraint problem solver.

12. The computer system for problem solving according to claim 10, wherein said memory portion further comprises software including a learning module for refining said expected problem solver behavior.

13. The computer system for problem solving according to claim 10, further comprising a problem transformer module for transforming said problem statement after receiving said problem statement from said input device.

14. The computer system for problem solving according to claim 10, further comprising a problem transformer module for transforming said problem solution before providing said problem solution to said output device.

15. The computer system for problem solving according to claim 10, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

16. The computer system for problem solving according to claim 15, further comprising a problem transformer module for transforming said problem

configuration parameters before providing said problem configuration parameters to said solving module.

17. The computer system for problem solving according to claim 10, wherein said software for determining expected solver behavior comprises a data structure, said data structure containing configuration parameters and expected behaviors for a plurality of problem types.

18. The computer system for problem solving according to claim 10, wherein said control computer comprises an embedded computer.

19. The computer system for problem solving according to claim 18, wherein said embedded computer system controls at least one operation within a copier or printer.

20. The computer system for problem solving according to claim 18, wherein said embedded computer system controls at least one operation within a process control system.

21. The computer system for problem solving according to claim 18, wherein said embedded computer system controls at least one operation within a diagnostics unit.

22. A computer system for problem solving, the system having implementation units communicating with the computer system, the system comprising:

an input device for providing the primary goal for the task to be performed;

a computer coupled to the output of said input device;

a memory portion coupled to said computer comprising:

a complexity module for configuring a problem statement and determining expected solver behavior;

a controllable solving module coupled to said complexity module for determining actual solver behavior;

a synthesis module for determining configuration parameter vectors;
and

comparison means for comparing said actual solver behavior with said expected solver behavior; and

output means for providing a statement of the problem solution.

23. The computer system for problem solving according to claim 22, wherein said problem solver comprises an adaptive constraint problem solver.

24. The computer system for problem solving according to claim 22, further comprising a learning module.

25. The computer system for problem solving according to claim 22, further comprising a problem transformer module for transforming said problem statement after receiving said problem statement from said input device.

26. The computer system for problem solving according to claim 22, further comprising a problem transformer module for transforming said problem solution before providing said problem solution to said output means.

27. The computer system for problem solving according to claim 22, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

28. The computer system for problem solving according to claim 27, further comprising a problem transformer module for transforming said problem configuration parameters before providing said problem configuration parameters to said solving module.

29. The computer system for problem solving according to claim 22, wherein said complexity module comprises a data structure, said data structure containing configuration parameters and expected behaviors for a plurality of problem types.

30. The computer system for problem solving according to claim 22, wherein said control computer comprises an embedded computer system.

31. The computer system for problem solving according to claim 30, wherein said embedded computer system controls at least one operation within a copier or printer.

32. The computer system for problem solving according to claim 30, wherein said embedded computer system controls at least one operation within a process control system.

33. The computer system for problem solving according to claim 30, wherein said embedded computer system controls at least one operation within a diagnostics unit.

34. A problem solver within a computer system, said problem solver comprising:

means for receiving a problem statement,

means for determining expected solver behavior associated with said problem statement;

means for providing configuration parameters for a plurality of problems;

means for determining a set of configuration parameter vectors;

means for performing a partial search with said configuration parameter vectors;

means for calculating actual solver behavior;

means for reviewing said actual solver behavior to determine if a problem solution has been found;

means for determining whether to perform a solver iteration step or to request an adaptation step if a problem solution has not been found;

means for performing a solver iteration step, comprising performing another search step, calculating a revised actual solver behavior and determining whether to repeat said solver iteration step;

means for comparing said actual solver behavior with said expected solver behavior;

means for requesting performance of an adaptation step;

means for performing an adaptation step, comprising modifying said configuration parameters, determining a revised expected problem solver behavior, and providing said modified configuration parameters and said revised expected problem solver behavior to said means for performing a solver iteration step; and

means for providing the problem solution to an output device.

35. A problem solving method for use in a computer system comprising the steps of:

receiving a problem statement;

configuring a problem solver with configuration parameters;

determining a set of configuration parameter vectors;

determining expected solver behavior associated with said configuration parameters for said problem statement;

searching for a solution with said configuration parameter vectors;

determining actual solver behavior;

determining if a problem solution has been found;

determining whether to perform a solving iteration step or an adaptation step if a problem solution has not been found;

performing said solver iteration step, when said solver iteration step is selected, comprising the steps of determining a new actual solver behavior and determining whether to repeat said iteration step;

repeating said solver iteration step until said adaptation step is selected;

comparing said actual solver behavior with said expected solver behavior when said adaptation step is selected;

performing said adaptation step, comprising the steps of modifying said configuration parameters, determining expected solver behavior associated with said modified configuration parameters, determining a revised actual solver behavior, reviewing said revised actual solver behavior to determine if a problem solution has been found, determining whether to perform said solver iteration step or to perform another adaptation step if a problem solution has not been found, and repeating said iteration step until said adaptation step is selected;

repeating said adaptation step until a problem solution is found; and
transmitting a solution statement.

36. The problem solving method according to claim 35, wherein said problem solving method comprises an adaptive constraint problem solving method.

37. The adaptive constraint problem solving method according to claim 35, further comprising the step of refining the control parameters.

38. The adaptive constraint problem solving method according to claim 35, further comprising the step of transforming said problem statement.

39. The adaptive constraint problem solving method according to claim 35, further comprising the step of transforming said problem solution.

40. The adaptive constraint problem solving method according to claim 35, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

41. The adaptive constraint problem solving method according to claim 40, further comprising the step of transforming said problem configuration parameters.

42. The adaptive constraint problem solving method according to claim 35, wherein the step of determining a set of configuration parameter vectors further comprises:

choosing a set of default configuration parameter vectors;

selecting an initial minimum point;

performing a local search;

evaluating actual behavior to determine whether to repeat a local search or select a different solver algorithm;

repeating a local search with a second minimum point when the step of repeating a local search is selected; and

revising the set of configuration parameter vectors for each search performed.

43. The adaptive constraint problem solving method according to claim 35, further comprising the step of selecting an algorithm for calculating a problem solution.

44. A problem solver stored via storage media, the storage media comprising:

a first plurality of binary values for receiving a problem statement transmission and storing the problem statement in a first data format;

a second plurality of binary values for transforming the first data format to a second data format;

a third plurality of binary values for determining expected solver behavior associated with said second data format;

a fourth plurality of binary values for selecting a set of configuration parameter vectors;

a fifth plurality of binary values for performing a partial search;

a sixth plurality of binary values for determining actual solver behavior associated with said second data format;

a seventh plurality of binary values for determining if a problem solution has been found;

an eighth plurality of binary values for determining whether to perform a solver iteration step or perform an adaptation step if a problem solution has not been found;

a ninth plurality of binary values for comparing said expected solver behavior and said actual solver behavior;

a tenth plurality of binary values for performing a solver iteration step;

an eleventh plurality of binary values for performing a solver adaptation step; and

a twelfth plurality of binary values for transmitting a solution statement in a third data format.